Persistent Sensitive Information Stored Unencrypted within the Memory Space of the Wickr iOS App

Published Date	TBD
Vendor	Wickr Inc.
Affected Versions	2.5.2 for iOS
Risk Rating	Medium
CVE Reference	TBD
Author	Sagi Shahar

Description

Wickr version 2.5.2 (iOS) was found to store user credentials persistently in the application's memory space. This may enable attackers to recover the user's credentials and hijack their Wickr account.

Cause

The application does not overwrite the memory that stores the sensitive information after it is dereferenced.

Impact

Successful exploitation may allow full access to the user's account.

Interim Workaround

None.

Technical Details

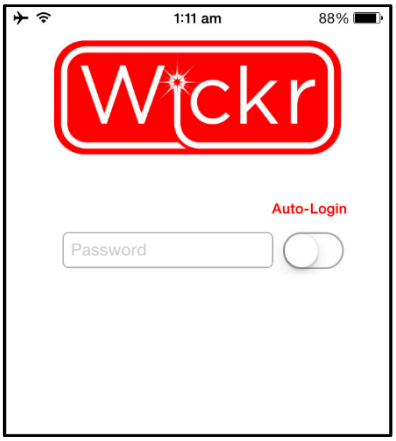
Wickr's authentication mechanism requires the user to input their password before gaining access to their sensitive information. It is assumed that once authentication is successful the password is no longer required for the application to function properly. This is thought to also be the case when the application enters the background as well as when the user is logged out completely.

While using the application it was observed that the password used for authentication remained in the application's memory space in clear-text.

The authentication view is controlled by the 'UserLogin' class, which contains various properties, one of them, for example, is:

- UITextField* passBox

The 'passBox' property is used by the 'UserLogin' view controller to store the password entered by the user to authenticate. The following screenshot shows the 'UserLogin' view and the 'passBox' text field:



After the user authenticates successfully, the application seems to dereference the 'UserLogin' view controller, however, the data that the object holds was not overwritten. By writing the heap memory space of the application into a file and extracting strings from the file it is possible to recover the clear-text password. This process holds effective also when the user has explicitly logged off from the application with the application running inactive in the background.

The following figure shows the process of writing the heap memory into files by using heapdump on a jail broken device:

```
asterisklabs:~/tools root# ./heapdump.sh Wickr
App Pid: 25819
GDB Version: 1708

mach-regions: 0x7d000 0x3c1000
mach-regions: 0x4a5000 0x4a7000
[snipped]
Dumping memory: Done
```

For Proof-of-Concept purposes, a string that is known to be part of the password was searched within the written files. The highlighted portion is the legitimate password:

```
asterisklabs:~/tools root# strings *.dmp | grep -i p4ss
p4ssw0rd
p4ssw0rd
p4ssw0rd
verylongp4ssw0rd
strings: object: dump0x1d09000.dmp truncated or malformed object (LC_SEGMENT
command 1 fileoff field plus filesize field extends past the end of the file)
strings: object: dump0x5131000.dmp truncated or malformed object (LC_SEGMENT
command 0 fileoff field plus filesize field extends past the end of the file)
strings: object: dump0x5148000.dmp truncated or malformed object (LC_SEGMENT
```

```
command 0 fileoff field plus filesize field extends past the end of the file)
strings: object: dump0x6558000.dmp truncated or malformed object (LC_SEGMENT
command 1 fileoff field extends past the end of the file)
strings: object: dump0x7d000.dmp truncated or malformed object (LC_SEGMENT
command 2 fileoff field plus filesize field extends past the end of the file)
```

To exploit this bug the phone needs to be jail broken and the application running in the background. However, due to the sensitive nature of the Wickr application and its strong built-in anti-forensic recovery features, this is believed to be a risk. In addition, the Wickr app does not do any jailbreak detection.

A more secure implementation can be implemented such that the password entered by the user is compared with its equivalent salted hash and not its clear-text version. Additionally, the application should utilise the 'NSMutableString' class that allows strings to be overwritten according to Apple's iOS API. As such, part of the application's dereference process should also include a procedure which overwrites the sensitive string with a random one.